Robust fusion of colour and depth data for RGB-D target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints

Xiao, Jingjing; Stolkin, Rustam; Gao, Yuqing; Leonardis, Ales

DOI:
10.1109/TCYB.2017.2740952

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Xiao, J, Stolkin, R, Gao, Y & Leonardis, A 2017, 'Robust fusion of colour and depth data for RGB-D target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints', *IEEE Transactions on Cybernetics*, vol. 99, pp. 1-15. https://doi.org/10.1109/TCYB.2017.2740952

Robust fusion of colour and depth data for RGB-D target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints

Jingjing Xiao, Rustam Stolkin, Yuqing Gao, Aleš Leonardis

***Abstract*—This paper presents a novel robust method for single target tracking in RGB-D images, and also contributes a substantial new benchmark dataset for evaluating RGB-D trackers. While a target object's colour distribution is reasonably motion-invariant, this is not true for the target's depth distribution, which continually varies as the target moves relative to the camera. It is therefore non-trivial to design target models which can fully exploit (potentially very rich) depth information for target tracking. For this reason, much of the previous RGB-D literature relies on colour information for tracking, while exploiting depth information only for occlusion reasoning. In contrast, we propose an adaptive range-invariant target depth model, and show how both depth and colour information can be fully and adaptively fused during the search for the target in each new RGB-D image. We introduce a new, hierarchical, two-layered target model (comprising local and global models) which uses spatio-temporal consistency constraints to achieve stable and robust on-the-fly target relearning. In the global layer, multiple features, derived from both colour and depth data, are adaptively fused to find a candidate target region. In ambiguous frames, where one or more features disagree, this global candidate region is further decomposed into smaller local candidate regions for matching to local-layer models of small target parts. We also note that conventional use of depth data, for occlusion reasoning, can easily trigger false occlusion detections when the target moves rapidly towards the camera. To overcome this problem, we show how combining target information with contextual information enables the target's depth constraint to be relaxed. Our adaptively relaxed depth constraints can robustly accommodate large and rapid target motion in the depth direction, while still enabling the use of depth data for highly accurate reasoning about occlusions. For evaluation, we introduce a new RGB-D benchmark dataset with per-frame annotated attributes and extensive bias analysis. Our tracker is evaluated using two different state-of-the-art methodologies, VOT [20] and OTB [45], and in both cases it significantly outperforms four other state-of-the-art RGB-D trackers from the literature.**

***Index Terms*—RGB-D tracking, range-invariant depth models, clustered decision tree**

I. Introduction

A. Motivation

Visual object tracking remains a challenging research problem, due to background clutter, occlusions, fast or erratic target motion, illumination changes, target scale changes and target deformations. In recent years, a variety of powerful new RGB trackers have been proposed, which demonstrate strong performance on challenging benchmark datasets. Well known examples include KCF [13], LGT [41], CDT [47], DMT [19], and very recent work such as DST [46], and ECM [49]. To further improve tracking performance, recent work has tried to incorporate additional features [23], [50], [43], [11], with depth information attracting growing interest [34]. However, while a target's RGB features are comparatively invariant to the target's motion, the target's depth information (by its very nature) will vary rapidly and significantly during target motion (especially motion towards or away from the camera). For this reason, many state-of-the-art RGB-D methods [7], [9], [28] rely mainly on RGB data for target tracking, and reserve depth information primarily for occlusion reasoning. However, in such methods, rapid target motion towards the camera can be easily mistaken for an occlusion, leading to tracking failure, Fig. 1. More details about the state-of-the-art trackers are provided in Sec. II.

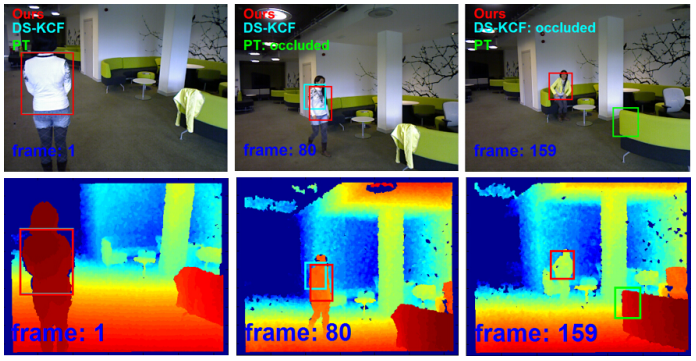


Fig. 1. RGB-D tracking of a target which exhibits large motion in the depth direction. Top and bottom images are matching pairs of RGB and D images, respectively. Red, blue and green bounding boxes represent our proposed tracker and the state-of-the-art trackers DS-KCF [6] and PT [34], respectively. In frame 80, following a rapid target depth change, PT [34] falsely reports "occluded" status (bounding box disappears). In frame 159, DS-KCF [6] falsely reports "occluded" status, while PT [34] has completely failed by drifting (becoming fixated on background clutter). Our proposed method (red bounding box) tracks successfully throughout.

B. Our novel RGB-D tracker

We address the tracking problem by proposing a new method for adaptively combining both RGB and depth information in a robust way during tracking. We propose a new,

J. Xiao is with Department of Medical Engineering, Xinqiao Hospital, Third Military Medical University, Chongqing, China. E-mail: shine636363@sina.com

R. Stolkin, Y. Gao and A. Leonardis are with the University of Birmingham, United Kingdom, E-mail: r.stolkin@cs.bham.ac.uk, yxg140@gmail.com, a.leonardis@cs.bham.ac.uk

hierarchical, two-layered (global-template layer and local-parts layer) RGB-D tracker which adaptively fuses a variety of features, derived from both the RGB and depth images. This two-layered target model is continually relearned on-the-fly (i.e., continuous adaptation of the targets that change their appearance during tracking). Both temporal and spatial consistency constraints are used to ensure that this continuous target model relearning is robust and stable.

In the global layer, a memory encodes temporal information, i.e., a history of global-layer feature values in previous frames, and this tracking memory is used to evaluate the consistency of newly extracted features from a candidate target region in the current frame. If one feature undergoes large temporal variation (i.e., low consistency), then a smaller weight is assigned to that feature modality. Thus the tracker adaptively weights in favour of the best feature modalities, while fusing multiple features for global target estimation.

If ambiguity is detected, during an attempt to match the global target model to a candidate image region, then this candidate region is further subdivided into a set of local regions, for matching to target parts within the local-layer of the target model. For robust matching, each target part is associated with multiple feature modalities, generated from both RGB and depth data. The statistics of each feature, in both the target region and a non-target region locally surrounding the target (which we refer to as the "context" region), are used to continuously re-evaluate the discriminating power of each feature and re-weight all features accordingly at each frame.

We also exploit a spatial constraint of the target with respect to depth values in regions surrounding the target, to adaptively estimate the range interval occupied by the target, thereby enforcing spatial consistency. This spatial constraint encodes common-sense or "simple physics" knowledge that target parts cannot be located behind background regions or in front of occluding objects.

C. Our new benchmark RGB-D dataset

Despite the availability of several high quality benchmark datasets for RGB tracker evaluation [1], [21], [24], [44], those available for RGB-D tracker evaluation are far more limited. Although the RGB-D tracking dataset PTB [34] made significant contributions, providing an early and pioneering step towards RGB-D benchmarking, we observe that this dataset suffers from a number of problems.

Firstly, it contains many sequences where the RGB and depth image pairs are significantly un-synchronised, e.g., Fig. 2. Unsynchronized image pairs cause ambiguities in annotated ground-truth. For example, given an RGB image and a depth (D) image which do not correspond, how should the human annotator annotate? Should he or she annotate according to the location of the target region in the RGB image, or the contradicting target region in the depth image? Such ambiguity in annotation directly leads to ambiguity in performance evaluation of the RGB-D tracking algorithm being tested.

Secondly, over half the PTB dataset [34] is devoted to human pedestrian tracking, which can bias the results of evaluating generic target trackers. In contrast, most of the current state-of-the-art tracking literature is aimed at creating "anything-trackers", i.e., the current state-of-the-art literature is not focused purely on pedestrian tracking, but addresses tracking of arbitrary objects with diverse sizes, shapes, appearances and motion behaviours.

Thirdly, the vast majority of benchmark videos in [34] are captured by a stationary camera. In contrast, the ability to handle camera motion is a major (and topical) visual tracking research challenge.

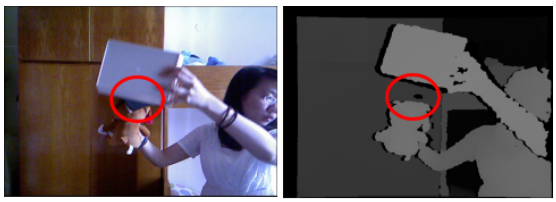
Fourthly, we notice that the dataset of [34] contains a very large number of videos, however many of these videos show very similar scenes, e.g., multiple videos of the same target object against the same background. Note that *large numbers of videos are not a sufficient condition to ensure a bias-free dataset*. Additionally, large numbers of videos result in extremely long and computationally intensive efforts when comparing and evaluating trackers. For example, repeatedly running several different trackers (especially with repeat runs to test various different parameter values, various degrees of initialisation error or other factors), can take many days or even weeks of computer run-time. In contrast, bias can be dramatically reduced by carefully selecting a *smaller* number of *more diverse* videos, resulting in improved performance evaluation, with the additional benefit of lower computational cost during performance evaluation.

While acknowledging the important and pioneering contribution of PTB [34], due to the above-mentioned issues we found it necessary to build a new RGB-D benchmark dataset, with correctly synchronised RGB and D channels, to evaluate our proposed RGB-D tracker in a consistent and unbiased way. We constructed our new RGB-D benchmark dataset with extensive bias analysis, and comparable size to the well-known and state-of-the-art benchmark RGB tracking dataset of the VOT challenge [1] (associated with the vision conferences ICCV/ECCV each year). Our RGB-D dataset comprises 36 video sequences, with each scene captured by both stationary and moving cameras. This is important, in order to distinguish whether tracking failures result from camera motion or from some other scene attribute (e.g., occlusion, clutter, or motion of the target object itself).

For every frame, in every video, we provide two categories of attribute annotation: binary and statistical. Three different binary ("yes" or "no") attributes were provided by human annotators (e.g., "Is this scene camouflaged, yes or no?"). Meanwhile seven different objective statistical attributes were also calculated: i.e., given human-annotated ground-truth bounding boxes for the target position in each frame, we objectively computed seven different statistical attributes, such as "degree of illumination variation" of the target image region during the video sequence. All of these attributes have been *per-frame* annotated to: i) ensure unbiased dataset construction; ii) aid functional understanding and explanations (e.g., "Is this tracker good or poor at handling occlusions?"); iii) assist end-users in choosing parameter values for RGB-D tracking algorithms, e.g., providing a dataset that can be used to help choose the best parameters for RGB-D tracking in a particular industrial or other practical application where a particular kind

of tracking attribute may be expected. Note that this *per-frame* annotation is very important, because most attributes (e.g., occlusion) only last for a small part of each video sequence, hence *per-sequence* annotation of such attributes (common in many datasets) is inadequate for fully analysing and understanding the performance of tracking algorithms with respect to different attributes.

We have evaluated our proposed RGB-D tracker using two different state-of-the-art evaluation methodologies: VOT [1] (re-initializes trackers after failures) and OTB [44] (without re-initialization). The results suggest that our proposed tracker significantly outperforms several other state-of-the-art RGB-D trackers [34], [6], [28] according to both evaluation methodologies.



Test sequence: ***bear_back*** (frame 24)

Test sequence: ***cup_book*** (frame 11)

Fig. 2. Some examples from Princeton RGB-D benchmark tracking dataset [34] which clearly show that RGB images and depth images are not synchronized.

D. Contributions and layout of this paper

The main contributions of this paper are: i) we introduce a new RGB-D tracking algorithm, which significantly outperforms several other state-of-the-art RGB-D trackers; ii) we introduce a substantial new benchmark RGB-D dataset (which is fully publicly available).

Our proposed tracker itself contains a variety of novel contributions: i) unlike many previous RGB-D trackers, we directly fuse RGB and depth information during matching of the target model to candidate image regions; ii) we continuously relearn the saliency of both depth and RGB features on-the-fly, and show how this can be accomplished using a new data fusion paradigm, by extending the concept of Clustered Decision Trees [47] which we previously developed for RGB tracking; iii) we show how both spatial and temporal consistency can be used in novel ways, to constrain the continuous relearning of targets which change their appearance, ensuring that such continuous model re-learning is stable and robust; iv) we show how common-sense or "simple physics" knowledge can be encoded in the form of a spatial constraint on the target's position with respect to both the background scene and occluding foreground clutter. This spatial constraint enables us to relax the motion constraints in the depth direction (very tightly constrained in most other RGB-D work), enabling us to handle large and rapid motions of the target in the depth direction, while simultaneously achieving very accurate occlusion reasoning.

The remainder of this paper is organised as follows. Related work is discussed in Sec. II. The proposed RGB-D tracker is described in Sec. III. Sec. IV presents our benchmark dataset. Sec. V presents experimental results. Sec. VI provides concluding remarks.

II. Related work

This section reviews recent work on both tracking algorithms and benchmark datasets.

A. Target representations for visual tracking

The choice of target representation is critical to tracking [16], with two main streams of research: global template trackers, e.g., [25], [27], [46]; and part-based trackers, e.g., [17], [31], [22]. Global representations reflect overall feature statistics of the target, simply and efficiently. However, global models can fail during significant target deformations or partial occlusions. In contrast, part-based methods can handle such problems more flexibly, albeit at an increased computational cost. Moreover, local parts can easily be distracted by cluttered scenes, causing part-based trackers to drift or fixate on target-like elements of background clutter. Furthermore, complex geometric constraints, from the relative motions of target parts, pose significant additional challenges.

Zhong et al. [52] and Cehovin et al. [41] proposed trackers which combined both global templates and local parts, and also combined information from multiple types of features. However, both methods fused multiple features in a homogeneous way, by simply multiplying the likelihoods of each feature to obtain an overall matching score. For such trackers, one poorly performing (e.g., camouflaged) feature can yield an erroneously high (or low) likelihood, which can destroy tracking performance if it is combined with the likelihoods of other features via a simple product rule. Therefore, it is essential to continuously re-evaluate the consistency and discriminative ability of each feature online (i.e., at each new frame), before performing adaptive feature fusion in an informed manner to achieve robust tracking.

B. Adaptive fusion of multiple feature modalities

Early work on continuously adaptive re-learning of the saliency of features, during online tracking of camouflaged targets against changing background scenes, was proposed in [35] for RGB images. Later work by Hong et al. [14] learned a discriminative metric that adaptively computed the importance of different features. Hu et al. [15] proposed a tracker which introduced a sparse weight constraint to dynamically select the relevant templates and a variance ratio measure to adaptively adjust the weights of different features. Posseger et al. [32] also proposed a distractor-aware target model to select salient colours in single target tracking. A unified model to select the best matching metric (attribution selection) and the most

stable sub-region of the target (spatial selection) for tracking was proposed in [18]. Our recent works, including [38], [48], [47], explored other methods for adaptive online re-weighting of feature saliency for robust tracking. In particular, this paper shows how our recent work, on Clustered Decision Trees [47] for adaptive RGB tracking, can be extended to provide a framework for efficiently and adaptively combining RGB and depth information to achieve robust RGB-D tracking.

C. Combining colour data with depth information

Recent works [9], [7] attempted to exploit depth information for tracking. However, these methods relied on RGB data for estimating the target's position in each 2D RGB image, while using depth information solely for reasoning about occlusions. A problem with such methods is that the target itself can easily become mislabelled as an occluding object, if it moves rapidly towards the camera. [29] proposed an RGB-D tracker using motion, colour appearance and HOG features where the corresponding likelihoods are simply multiplied together. The method can effectively track the pedestrians within groups, however, the assumption of a static camera limits its application scenarios. [26] proposed a boosting approach using three types of RGB-D features, i.e., Haar-like features, colour-based HOG and depth-based HOG features. However, the work is specifically targeted at pedestrian tracking, and not at tracking arbitrary targets which is one of our main objectives.

A sophisticated RGB-D tracking work was carried out by Song et al. [34]. They proposed several different RGB-D trackers, using both 2D and 3D models, to evaluate the extent to which the addition of depth information could improve overall tracking performance. In [34], the best tracking performance was achieved by using an RGB-D HOG feature detector and a point cloud feature detector. The detection results of each feature were summed, and then adjusted using a 3D iterative closest point (ICP) algorithm for refining the final estimate of the target position. [34] also proposed that occlusions could be detected when the target region depth histogram develops a newly rising peak with a smaller depth value than the target. However, this heuristic only works when the target estimation is highly accurate and the depth information is relatively stable, which is not the case in situations where the target moves rapidly towards the camera.

[6] applied kernel correlation filters in RGB and depth maps respectively, where sudden changes in depth values were used to infer occlusion situations. However, this global tracker has difficulty handling significant target deformations (see the above discussion about the difficulties of global target models for such circumstances). In [12], the authors further enhanced the work of [6] by handling the shape deformations using the estimated target depth distribution and a segmentation mask. However, both trackers rely solely on depth data for occlusion reasoning, which can cause failures, as shown in Fig. 1. Meshgia et al. [28] proposed an "occlusion aware" particle filter to handle complex and persistent occlusions. However, this tracker uses a tightly constrained, pre-defined depth threshold. It therefore has difficulty tracking targets which exhibit large ranges of motion in the depth direction. Bibi et al. [5] proposed 3D part-based tracker with automatic synchronization and registration. However, they judge the occlusion in terms of the thresholds computed in the first frame without updating, which fails in cases of significant and fast inwards/outwards movements.

Unlike the above-mentioned works [34], [6], [51], [28], our tracker fuses both RGB and depth data during all aspects of tracking, and does not reserve depth data merely for occlusion reasoning. Specifically, we: i) extend our RGB tracker [47] to combine both RGB and depth data in a two-layered model with Clustered Decision Trees; ii) exploit both temporal and spatial consistency constraints to adaptively fuse RGB-D features; and iii) propose a spatial constraint on the target's position with respect to the depth context, which enables relaxation of motion constraints in the depth direction while still maintaining robust reasoning about occlusions.

Note that this work is not merely an incremental extension of our previous CVPR paper [47], on RGB tracking using Clustered Decision Trees, to also handle depth data. Unlike the target model and matching scheme proposed in [47], this paper presents a new model which: i) fuses multiple features within the top layer (in contrast, our previous method [47] required a separate layer for each feature); ii) achieves greater computational efficiency by being hierarchical, in that it uses only the global target model during unambiguous (high confidence) situations, but automatically triggers the use of the local-parts layer if ambiguity is detected in the global layer; iii) introduces a new method for robustly matching target models to candidate image regions, by combining the likelihoods of local-parts, the global-model, and contextual information in a cross-constrained paradigm; iv) proposes a spatial constraint of the target with respect to other objects in the depth context, to adaptively estimate the range interval occupied by the target, enabling continuous on-the-fly relearning of a range-invariant target depth model.

D. Performance evaluation methodologies and benchmarks for target tracking

For performance evaluation and comparisons of visual tracking algorithms, a variety of benchmark ground-truthed video datasets have been created. Perhaps the earliest published work on creating fully ground-truthed visual tracking benchmark datasets was that of [36], [37], which generated videos in various kinds of good and poor visibility conditions (variable lighting, smoke/fog etc.), for which ground-truth 3D poses of tracked objects (rigid bodies) relative to the camera were measured for every frame. More modern and better known tracking benchmark datasets include ALOV++ [33], OTB [45], NUS-PRO [24] for RGB tracking, and PTB [34] for RGB-D tracking. However, a common problem of the above datasets is that they either lack attributes annotation entirely, or their attributes are only *per-sequence* annotated rather than *per-frame* annotated, and all annotations are assigned based on the intuition of human annotators. This makes it difficult to accurately categorise a tracker's performance with respect to different attributes or tracking conditions. For example, per-sequence annotated attributes (e.g., "occlusion") do not last

throughout the entire sequence [21]. It is therefore preferable to evaluate each tracker with respect to attributes annotated on a *per-frame* basis, for a more accurate and meaningful analysis.

For several years, Kristan et al. [1] have been organising the well known Visual Object Tracking (VOT) challenge, held each year at a top computer vision conference (ICCV or ECCV). The pioneering VOT benchmark dataset is based on per-frame annotation, however, this dataset is annotated only with binary attribute labels according to the intuition of human annotators (e.g., "Does this image contain clutter? Yes or no?"). Additionally, the VOT dataset is limited to 2D RGB videos (later extended to also include some 2D infra-red videos), and does not include any RGB-D data.

Since a per-frame annotated RGB-D tracking dataset is currently unavailable, we have created a new RGB-D benchmark dataset which: i) ensures maximum diversity with minimum size/cost through a rigorous bias analysis; ii) incorporates complete *per-frame* annotation for every frame in every sequence; iii) includes human annotation of three binary attributes, as well as statistical evaluation of seven more objective numerical attributes, for every frame; iv) includes videos from both moving and stationary cameras for every scene, which is necessary for disambiguating failures caused by camera motion from failures caused by other kinds of video sequence attributes.

III. Proposed RGB-D tracker

Our proposed RGB-D tracker represents the target as a two-layer model, comprising a "global" layer representing the overall target object, and a "local" layer which represents the target as a collection of smaller target parts. At each new frame, our tracker begins by propagating the target's bounding box to multiple candidate target regions in the new frame, and then using the global layer of the target model to provide an initial evaluation of the likelihood of each such candidate target location. Likelihoods are evaluated by combining the opinions of multiple features. The weights of each feature's opinion are adaptively re-weighted on-the-fly, by continuously relearning models of the temporal consistency of each feature.

If these initial likelihood evaluations, at the global layer, are deemed "ambiguous" (defined later), then the tracker progresses to the local layer, which attempts to match smaller parts of the target to smaller sub-regions within each candidate bounding box. At each frame, the tracker continually re-evaluates the saliency of each target part by comparing feature statistics of the candidate target region and a local image region surrounding the target (which we refer to as the "context" region), illustrated in Fig. 3.

The following sections describe: global tracking based on temporal consistency, in Sec. III-A; and part-based tracking with spatial context constraint, in Sec. III-B.

A. Global target tracking by exploiting temporal consistency

The tracker first propagates a set of samples, to find candidate target regions in both RGB and depth images. Each such candidate region is then evaluated, by both colour and depth-based features, in order to select the most likely candidate region. In this sense, "likely" candidates are those that yield

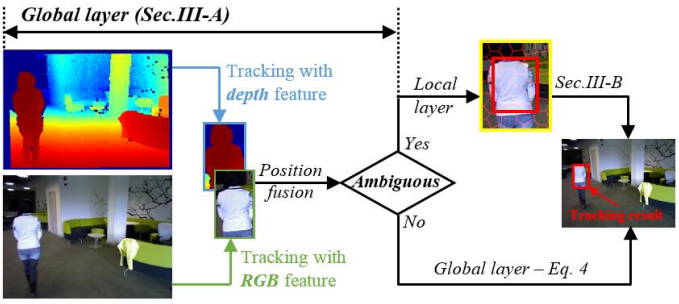


Fig. 3. A diagram depicting the proposed method. In the global layer, the algorithm utilises two different features, i.e., RGB and depth, to provide candidate target positions. When the positions provided by the global layer are ambiguous, the tracker progresses to the local layer, matching smaller parts of the target to the smaller sub-regions within each candidate's bounding box. In the local layer, the red bounding box indicates the target region $\mathcal{O}$, as estimated by the global layer, and the region between the red bounding box and the yellow bounding box indicates context $\mathcal{C}$.

a high matching score with the global target model, for the respective features.

Different features may agree or disagree about the best candidate target locations, and the amount of disagreement may vary. In "unambiguous" situations (where all features are in strong agreement as to the target location), overall target localisation can be rapidly and cheaply resolved within the global layer of the tracker. In contrast, in "ambiguous" situations (where different features disagree within the global layer), our tracker then invokes its local layer, to seek finer matching at the scale of smaller target parts.

Note that, even in unambiguous situations which are resolved at the global layer, the fusion of multiple features to determine an overall target location remains non-trivial. Even in these "unambiguous" situations, we robustly fuse the opinions of each feature modality in an adaptive manner, by continuously relearning regression models of each feature's temporal consistency.

1) Global-layer features: In the global layer, tracking is based on the kernelised correlation filter (KCF) which measures the similarity between the image template and the image candidate region to produce the correlation peaks for the target regions while low response for the background [13]. Correlation filters take advantage of the fact that the correlation operations of two image regions could be conducted by element-wise multiplications using Discrete Fourier Transform, and have proved to be competitively efficient at hundreds of frames-per-second. Therefore, we propose to use individual features in RGB and depth images separately in kernelised correlation filter to find the target position in the global layer. In RGB images, we extract colour attributes [40] defined as linguistic colour labels with eleven basic colour terms, which have been shown to perform well in object detection, tracking and recognition [10]. In depth images, we use depth-HOG as the global depth feature. We denote the target estimate, found by colour KCF at the k^{th} frame, as $\mathbf{b}_{rgb,k}$, with its corresponding likelihood being $p(\mathbf{b}_{rgb,k})$. Similarly, we denote the target estimate, found by depth KCF at the k^{th} frame, as $\mathbf{b}_{d,k}$ with likelihood $p(\mathbf{b}_{d,k})$.

2) Ambiguous and unambiguous tracking situations: Based on the estimated target bounding boxes, $\mathbf{b}_{rgb,k}$ and $\mathbf{b}_{d,k}$, we distinguish two different tracking situations: "ambiguous" and "unambiguous". A high overlap ratio between the set of pixels $\mathcal{S}(\mathbf{b}_{rgb,k})$ and $\mathcal{S}(\mathbf{b}_{d,k})$ in the two bounding boxes indicates an unambiguous tracking situation [42], whereas a low overlap ratio indicates an ambiguous situation. Specifically, the situation is unambiguous if:

$$\mathcal{S}(\mathbf{b}_{rgb,k}) \cap \mathcal{S}(\mathbf{b}_{d,k}) > \lambda_{\Delta}\mathcal{O} \tag{1}$$

where $\mathcal{O}$ is the magnitude of the target's image region (i.e., target bounding box size), and λ_{Δ} is a threshold for determining ambiguity between $\mathbf{b}_{rgb,k}$ and $\mathbf{b}_{d,k}$. In practice we find that a value of $\lambda_{\Delta} = 0.9$ works well to avoid the tracker failing by drifting. This value of λ_{Δ} is informed by our observation that global trackers tend to begin failing when the amount of overlap between the true and estimated target regions falls below $\lambda_{\Delta} = 0.9$.

3) Online learning of likelihood regression models for each feature: In unambiguous situations, the likelihoods of each feature, $p(\mathbf{b}_{rgb,k})$ and $p(\mathbf{b}_{d,k})$, are used to update regression models. These likelihood regression models can then be used to compute a "temporal consistency measure" for each feature. These temporal consistencies can then be used to robustly (i.e., adaptively) fuse the opinions of both the colour and depth features in order to decide on an overall estimate of the target's location in the current frame.

The temporal history of a feature's matching scores in unambiguous frames is denoted as $\mathbf{F}_{rgb} = \{f_{rgb,1}, .., f_{rgb,k-1}\}$. Since the tracking scenario can change dramatically, the data in the early frames might be outdated and not suitable for model estimation. Therefore, to better predict the changes of the features, we use the data in the latest five frames where linear regression is sufficiently good to fit the data. Those data is used to continuously re-learn a linear regression model $\hat{f}_{rgb,k} = \alpha k + \alpha_0$, i.e., α and α_0 are continuously relearned online. During *ambiguous* situations, training of the regression model is temporarily switched off to prevent erroneous model re-learning.

4) Global layer matching procedure: We first describe the matching procedure in the RGB image and denote its matching score as $f_{rgb,k} = p(\mathbf{b}_{rgb,k})$. The difference between the feature matching score in the current frame $f_{rgb,k}$ and a predicted score $\hat{f}_{rgb,k}$ from the regression model, is used to measure temporal consistency. Feature weights are assigned by Eq. 2 where a feature with higher consistency score will be assigned a higher weight.

$$p(f_{rgb,k}|\mathbf{F}_{rgb}) = \exp(-|f_{rgb,k} - \hat{f}_{rgb,k}|) \tag{2}$$

In depth images, $p(f_{d,k}|\mathbf{F}_d)$ is obtained in a similar way. The final weights for fusion are computed by considering the temporal consistency of both RGB and depth features:

$$\begin{cases} \hat{p}(\mathbf{b}_{rgb,k}) = \frac{p(f_{rgb,k}|\mathbf{F}_{rgb})}{p(f_{rgb,k}|\mathbf{F}_{rgb}) + p(f_{d,k}|\mathbf{F}_d)}, \\ \hat{p}(\mathbf{b}_{d,k}) = \frac{p(f_{d,k}|\mathbf{F}_d)}{p(f_{rgb,k}|\mathbf{F}_{rgb}) + p(f_{d,k}|\mathbf{F}_d)}. \end{cases} \\ \text{if} \quad \mathcal{S}(\mathbf{b}_{rgb,k}) \cap \mathcal{S}(\mathbf{b}_{d,k}) > \lambda_{\Delta}\mathcal{O} \tag{3}$$

5) Target position estimation: The final target position estimate $\mathbf{b}_k$ for the global layer can now be conveniently obtained as a weighted linear combination of $\mathbf{b}_{rgb,k}$ and $\mathbf{b}_{d,k}$:

$$\mathbf{b}_k = \hat{p}(\mathbf{b}_{rgb,k})\mathbf{b}_{rgb,k} + \hat{p}(\mathbf{b}_{d,k})\mathbf{b}_{d,k} \tag{4}$$

Large differences between $\mathbf{b}_{rgb,k}$ and $\mathbf{b}_{d,k}$ indicate *ambiguous* situations, where at least one feature is likely erroneous.

B. Local tracking of target parts by exploiting spatial context

If an "ambiguous" situation is detected by the tracker's global layer, then the algorithm will progress to the local layer for more accurate tracking. The target's bounding box region $\mathcal{O}$ (roughly estimated by the tracker's global layer) is first enlarged to include a context region $\mathcal{C}$, which is a ring-shaped region surrounding the target bounding box region, as shown in Fig. 3. This gives a total search region:

$$\Omega = \mathcal{O} \cup \mathcal{C} \tag{5}$$

where Ω is a region of the image which we will search, to try and find matches to small parts of the target. To do this, the search region Ω is broken down into many small candidate image sub-regions $\mathcal{R}$. Matching of the local layer of the target model is then a problem of finding a candidate sub-region $\mathcal{R}(j)$ which best matches a target part i, for all parts.

Note: in contrast to many other part-based methods, which use a simple square grid to break down search regions into small candidate parts regions, we instead do this by using the super-pixel segmentation method of [3]. Super-pixels are strong candidates for matching to target parts, because each super-pixel is relatively homogeneous. In contrast, random or grid breakdowns of images into sub-regions, are more likely to result in a single target part being divided between two such sub-regions, which then leads to parts-level matching errors.

1) Three matching metrics for target parts: To match target parts to local image regions, we use three different matching metrics which take into account: local target parts information; global target information; and contextual information (i.e., information from the image region surrounding the expected target location).

Local colour based likelihood of target part i is defined as the similarity between the part's colour histogram $\mathbf{c}_t^i$ and the colour histogram $\mathbf{c}_c^j$ of a local candidate image region $\mathcal{R}(j)$. To match the i^{th} target part, local candidate regions $\mathcal{R}(j)$ are searched to find the best-matching region:

$$\hat{j} = arg\max\ p(i,j), \quad p(i,j) = \mathcal{B}(\mathbf{c}_t^i, \mathbf{c}_c^j) \\ \text{s.t. } p(i,j) > \lambda_{rgb} p(\mathbf{f}_{rgb,k}|\mathbf{F}_{rgb}) \tag{6}$$

where $\hat{j}$ denotes the best match out of local image regions $\mathcal{R}(j)$; $p(i,j)$ represents the local colour likelihood; $\mathcal{B}(,)$ is the Bhattacharayya similarity metric [4]; λ_{rgb} is a scale factor. If the matching likelihood $p(i,\hat{j})$, between part i and the best candidate $\hat{j}$, falls below the (continuously relearned) threshold $\lambda_{rgb} p(\mathbf{f}_{rgb,k}|\mathbf{F}_{rgb})$, then that target part is flagged as being occluded.

Note that much of the RGB tracking literature, e.g., [30], [47], used a simple heuristic for detecting occlusions: an occlusion was flagged whenever the target matching likelihood

dropped below a pre-defined minimum likelihood threshold. Unfortunately, this simple approach can fail under variable tracking conditions, e.g., during illumination changes. Therefore, we instead use the temporal consistency-based feature weight $p(\mathbf{f}_{rgb,k}|\mathbf{F}_{rgb})$ derived in Eq. 2, to continuously relearn the constraint in Eq. 6, which adaptively (and therefore more robustly) detects occluded parts of the target.

Global colour based likelihood is used to distinguish a pixel within the target region from surrounding (contextual) background pixels in RGB images [32], while suppressing distraction from background clutter.

Let $\mathcal{I}_{rgb,k}(x)$ be the colour of a pixel at position x in the k^{th} image frame $\mathcal{I}_{rgb,k}$. Given some region of the image, $\mathbf{h}_{rgb,k}(b^x_{rgb})$ is the probability returned by bin b^x_{rgb} of the RGB histogram computed from all pixels of that region.

Given a search region Ω, comprising an estimated target region $\mathcal{O}$ and surrounding context region $\mathcal{C}$, Bayes law yields the global colour-based likelihood of pixel x as:

$$p(x|\mathcal{O},\mathcal{C},b^x_{rgb}) \approx \frac{p(b^x_{rgb}|x\in\mathcal{O})p(x\in\mathcal{O})}{\sum\limits_{\forall x\in\Omega=\mathcal{O}\cup\mathcal{C}} p(b^x_{rgb}|x\in\Omega)p(x\in\Omega)} \quad (7)$$

where $p(b^x_{rgb}|x\in\mathcal{O}) \approx \mathbf{h}^{\mathcal{O}}_{rgb,k}(b^x_{rgb})$ while $p(x\in\mathcal{O}) \approx \frac{|\mathcal{O}|}{|\mathcal{O}|+|\mathcal{C}|}$ where $|.|$ represents the region size, as first proposed in [35].

Based on the above method for finding the global likelihood of an individual pixel, we now define the *global* colour-based likelihood $p_{rgb}(j)$ of a *local* candidate region $\mathcal{R}(j)$ as the sum of the likelihoods of the pixels contained in the region:

$$p(\mathcal{R}(j)) = \sum_{x\in\mathcal{R}(j)} p(x|\mathcal{O},\mathcal{C},b^x_{rgb}) \quad (8)$$

We can now search region Ω to find a sub-region $\mathcal{R}(j)$ which best matches target part i as:

$$\hat{j} = arg\max \sum_{x\in\mathcal{R}(j)} p(x|\mathcal{O},\mathcal{C},b^x_{rgb}) \quad (9)$$

Global depth based likelihood is obtained from a special kind of target depth histogram, similar to the way in which the global colour-based likelihood is obtained from the target colour histogram. Note that target depth can change relatively fast as the target moves. Therefore we propose a continuously relearned depth histogram target model, which is "position-shifted" to enable a range-invariant model of the target's depth data.

To continuously relearn this range-invariant target depth model, we exploit spatial constraints on the motions of the target with respect to the depth context. These constraints encode "simple physics" or common sense knowledge, that a target part cannot be deeper than a background region, and also cannot be less deep than an occluding object.

Defining a narrow depth interval for the target may fail to handle target motion in the depth direction (equivalent to over-fitting with respect to depth). Using too wide an interval could be regarded as under-fitting, and risks conflating the target and background information causing matching difficulties. To overcome these issues, we initially compute the foreground and background depth constraints from depth histograms of both target and context separately, Fig. 4.

We first seek a depth interval in which the target might be located at the current frame k. [6], [34] computed this depth interval solely from the observed target depth histogram in frame $k-1$, which tightly constrains the target, Fig. 4, to lie between target foreground and background constraints ($D^f_{T,k-1}$ and $D^b_{T,k-1}$). Unfortunately, using the range interval of frame $k-1$ to estimate occlusions in frame k, will fail when the target undergoes rapid motion in the depth direction, Fig. 1. Instead, we propose a spatial constraint on the target pose with respect to depth clutter, which allows us to relax the constraints of ($D^f_{T,k-1}$ and $D^b_{T,k-1}$), enabling the proposed tracker to robustly handle large target depth variations.

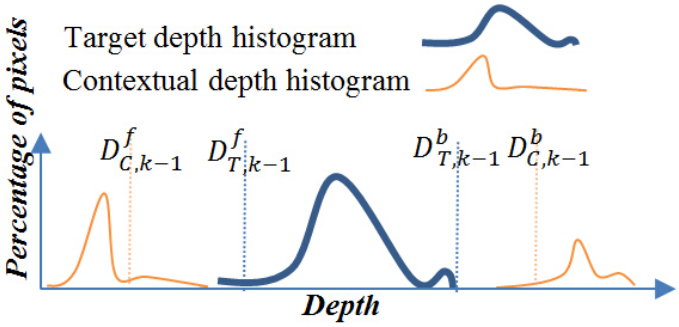


Fig. 4. Depth histogram. $D^f_{T,k-1}$ and $D^b_{T,k-1}$ represent foreground and background target constraints respectively, computed from the target depth histogram. Constraints $D^f_{C,k-1}$ and $D^b_{C,k-1}$ are computed from the contextual depth histogram.

We denote the observed target and contextual depth histograms at the $k-1$th frame as $\mathbf{h}^T_{d,k-1}$ and $\mathbf{h}^C_{d,k-1}$ respectively. Between constraints $D^f_{T,k-1}$ and $D^b_{T,k-1}$ of the target's depth histogram, is a narrow interval which we hope should contain a large proportion of target pixels. Between the context histogram constraints, $D^f_{C,k-1}$ and $D^b_{C,k-1}$, is a (typically) much larger interval which we hope contains a comparatively small proportion of target pixels. We define information loss (proportion of pixels missing from a depth interval) as:

$$\begin{aligned} L^T_{d,k-1} &= 1 - \textstyle\sum_{b_d=D^f_{T,k-1}}^{D^b_{T,k-1}} \mathbf{h}^T_{d,k-1}(b_d) \\ L^C_{d,k-1} &= 1 - \textstyle\sum_{b_d=D^f_{C,k-1}}^{D^b_{C,k-1}} \mathbf{h}^C_{d,k-1}(b_d) \end{aligned} \quad (10)$$

where $L^T_{d,k-1}$ and $L^C_{d,k-1}$ denote information loss for depth intervals obtained from the target $\mathbf{h}^T_{d,k-1}$ and contextual $\mathbf{h}^C_{d,k-1}$ depth histograms respectively, where b_d denotes a bin of a depth histogram.

We seek an optimal depth constraint interval, which will *minimize* target information loss, while *maximizing* contextual information loss, computed by:

$$arg \min_{(\hat{D}^f_{T,k-1},\hat{D}^b_{T,k-1})} \begin{cases} L^T_{d,k-1}, \\ D^{bf}_{T,k-1}. \end{cases} \text{s.t.} \begin{cases} L^T_{d,k-1} < \lambda_d, \\ D^{bf}_{T,k-1} > 0. \end{cases} \quad (11)$$

$$arg \max_{(\hat{D}^f_{C,k-1},\hat{D}^b_{C,k-1})} \begin{cases} L^C_{d,k-1}, \\ D^{bf}_{C,k}. \end{cases} \text{s.t.} \begin{cases} L^C_{d,k-1} > 1-\lambda_d, \\ D^{bf}_{C,k} > 0. \end{cases} \quad (12)$$

where $D^{bf}_{C,k} = D^{b}_{C,k} - D^{f}_{C,k}$ and $D^{bf}_{T,k-1} = D^{b}_{T,k-1} - D^{f}_{T,k-1}$. λ_d is a threshold parameter to ensure that the target depth interval contains sufficient pixels. Eq. 11 and Eq. 12 constitute a multi-objective optimization problem which we solve by combinatorial search, to yield an *adaptive depth interval*:

$$\begin{aligned} D^{f}_{A,k-1} &= \frac{D^{f}_{T,k-1} + \sum_{\phi\in\{b,f\}} \mathcal{H}(D^{f}_{T,k-1} - D^{\phi}_{C,k-1}) D^{\phi}_{C,k-1}}{1 + \sum_{\phi\in\{b,f\}} \mathcal{H}(D^{f}_{T,k-1} - D^{\phi}_{C,k-1})} \\ &\quad s.t. \quad D^{f}_{T,k-1} > D^{f}_{C,k-1} \\ D^{b}_{A,k-1} &= \frac{D^{b}_{T,k-1} + \sum_{\phi\in\{b,f\}} \mathcal{H}(D^{b}_{T,k-1} - D^{\phi}_{C,k-1}) D^{\phi}_{C,k-1}}{1 + \sum_{\phi\in\{b,f\}} \mathcal{H}(D^{b}_{T,k-1} - D^{\phi}_{C,k-1})} \\ &\quad s.t. \quad D^{b}_{T,k-1} < D^{b}_{C,k-1} \end{aligned} \tag{13}$$

where $\mathcal{H}$ is the Heaviside step function. This ensures that, even if the context-based depth interval locates completely on one side of the target-based depth interval, both constraints can still be combined to generate an overall *adaptive depth interval*, defined by $D^{b}_{A,k-1}$ and $D^{f}_{A,k-1}$. This adaptive depth interval achieves robustness by encoding common-sense physical knowledge, that targets cannot recede behind background objects, and also cannot approach closer to the camera than occluding objects. If these common-sense conditions are not met ($D^{f}_{T,k-1} \leqslant D^{f}_{C,k-1}$ or $D^{b}_{T,k-1} \geqslant D^{b}_{C,k-1}$) then an *ambiguity* situation in the depth modality is detected and the corresponding constraint becomes invalid.

In frame k, we regard any global layer RGB-D pixels (those inside the 2D image bounding box), which *also lie within the adaptive depth interval*, as belonging to the target. The mean depth $\bar{D}_k$ of those target pixels is now used to *position-shift* the target depth histogram to the new expected target position:

$$\hat{\mathbf{h}}^{T}_{d,k}(b^{x}_{d}) = \mathbf{h}^{T}_{d,k-1}(b^{x}_{d} + \bar{D}_k - \bar{D}_{k-1}) \tag{14}$$

where $\mathbf{h}^{T}_{d,k-1}$ is the target depth histogram observed at frame $k-1$. $b^{x}_{d} = \mathcal{I}_{d,k}(x)$ is the b^{x}_{d}-th bin in the depth histogram, corresponding to pixel x in image $\mathcal{I}_{d,k}$. $\bar{D}_{k-1}$ is the mean depth of target pixels at frame $k-1$.

Note that a conventional target depth histogram cannot be used for tracking (e.g., in the same way as a colour histogram), because it is not a motion-invariant target feature. By position-shifting with respect to expected target motion, we have effectively created a *range-invariant* target depth histogram, $\mathbf{h}^{T}_{d,k}(b^{x}_{d})$, which we can then use for tracking by generating likelihoods for target parts.

Next, the position-shifted target *depth* histogram in Eq. 14 and the contextual *depth* histogram $\mathbf{h}^{C}_{d,k-1}$ are used in a similar way as the target and context *colour* histograms are used in Eq. 7 and Eq. 9, to compute a global depth-based likelihood $p_d(i,j)$ which can be used for matching a candidate image sub-region $\mathcal{R}(j)$ to a target part i according to the criterion:

$$\begin{aligned} \hat{j} &= arg\max \textstyle\sum_{x\in\mathcal{R}(j)} p(x|\mathcal{O},\mathcal{C},b^{x}_{d}) \\ \text{s.t.} \quad D^{f}_{A,k-1} &< \frac{\sum_{x\in\mathcal{R}(j)} \mathcal{I}_{d,k}(x)}{\sum_{x\in\mathcal{R}(j)}} < D^{b}_{A,k-1} \end{aligned} \tag{15}$$

2) Matching of a target part to a candidate image region: Our tracker applies the three matching metrics (local colour, global colour and depth likelihoods, Eq. 6, Eq. 9 and Eq. 15) to three levels of a Clustered Decision Tree. We previously proposed the concept of Clustered Decision Trees, in our CVPR'15 paper [47], as a method for robust, but computationally efficient, matching of target parts i to local image regions j. The Clustered Decision Tree initially attempts to match a part using a single feature (first level on the tree), and then progresses to additional features (deeper levels of the tree), if the initial feature (first level) yields several possible candidate image regions (a "cluster" of regions) which share similar likelihoods to the best candidate j_{best}:

$$\mathbb{C}_{m,k}(j) = \begin{cases} 1, & \text{if} \quad |p^{j}_{m,k} - p^{j_{best}}_{m,k}| \leq \sigma_{m,k} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where $p^{j}_{m,k}$ is the likelihood of candidate local region j computed by matching metric m and $\sigma_{m,k}$ is the corresponding standard deviation. $\mathbb{C}_{m,k}(j)$ denotes whether or not region j is added to the cluster. For each target part, if the matching likelihood of any region j is sufficiently close to that of the best matching region j_{best}, then $\mathbb{C}_{m,k}(j)$ is set to 1, and region j is clustered with region j_{best}. $\mathbb{C}_{m,k}(j) = 1$ indicates that the current level of the decision tree (corresponding to use of a particular feature) has failed to confidently match the target part to a unique image region.

The tree then progresses to its next level (using an additional feature), to try and disambiguate candidate regions that remain clustered. When the target part finds a unique matching candidate region ($\sigma_{m,k}$ becomes zero), then the clustered decision tree algorithm terminates. If no successful matching region can be found, then that target part is regarded as occluded. The final target position is computed from the distribution of all matched local parts. Each local part, associated with a clustered decision tree, also contributes to a strong classifier for occlusion reasoning, by computing the ratio of matched local parts:

$$\mathbb{V}_k = N_{matched,k}/N_{all} \tag{17}$$

where $N_{matched,k}$ is the number of matched local parts and N_{all} is the number of all target parts. $\mathbb{V}_k$ can then be used as a "visibility" metric. When $\mathbb{V}_k$ becomes zero, then the target is considered to be fully occluded and its status will remain the same without updating. In the next frame, the algorithm will search for the target following the aforementioned procedure.

C. Model updating

It is crucial to update the target model to accommodate appearance changes, learn newly appearing information, and delete old information. Our proposed tracker does this via two mechanisms: adapting the old local and global layer target models - as described earlier in section III of this paper; and adding new models of new local layer target parts derived from super-pixels, using the mechanism explained in our previous RGB tracking work [47].

IV. Benchmark Dataset

Due to the limitations of currently available RGB-D benchmarks (see discussion in Sec. I), we have created a new dataset to evaluate our proposed method. Our dataset comprises 36 video sequences with an average of 300 frames per sequence,

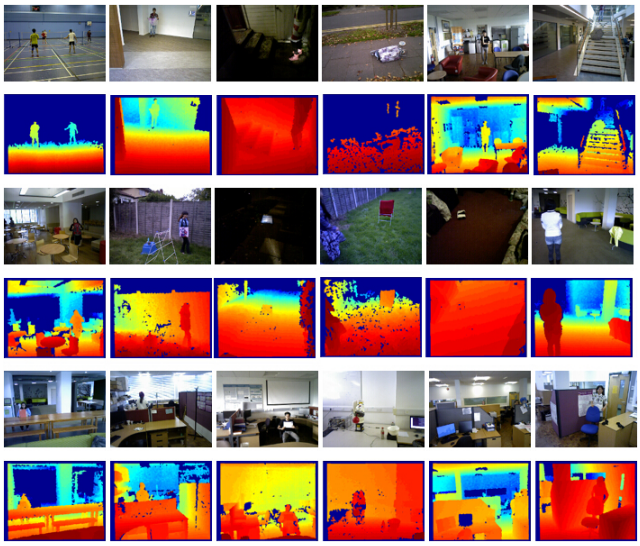


Fig. 5. The constructed RGB-D tracking dataset with 36 video sequences.The first, third, and fifth row depict the captured scenes in RGB images, and the second, fourth, and sixth row show the corresponding depth images.

and maximum sequence length of 700 frames. Some example scenes from the dataset are shown in Fig. 5 [1].

A. Dataset construction

1) Hardware set-up: For depth images, Microsoft Kinect and Asus Xtion are similarly accurate for close ranges up to 3.5m. However, Kinect is relatively bulky (12" x 3" x 2.5") and heavy (3.0 lb), and it requires an additional AC-DC power supply. In contrast, Asus Xtion is more compact (7" x 2" x 1.5") and lighter (0.5 lb) and does not need an additional power supply. This makes Asus Xtion more convenient for our dataset construction. Therefore, two ASUS Xtion RGB-D sensors were used to simultaneously record each scene. Since motion of the RGB-D camera often causes significant depth variation, video sequences recorded by moving RGB-D cameras engender greater tracking challenges, as compared to sequences recorded by stationary cameras. Therefore, for each scene, one sensor was fixed and the other sensor was moved continuously, along arbitrary trajectories, to evaluate the performance of trackers under conditions of arbitrary camera motion. Most sequences were recorded indoors with target depth ranging from 0.5 to 8 meters. To ensure the diversity of the scenes, and to investigate extreme tracking circumstances [2], some outdoor scenes were collected during the night. Even with the OpenNI2 synchronous function, RGB images and depth images are not always well synchronised, similar to problems observed in the pioneering dataset of PTB [34]. Therefore, time stamps were used to drop asynchronous image pairs.

2) Bounding box annotation: Human annotators were asked to manually select the minimum size of bounding box which fully contains the target object. Because our dataset is specifically designed for testing target tracking algorithms (not for testing object detection algorithms), the target objects were always fully or partially visible in every frame. In the case of partial occlusion, annotators were instructed to ensure that the ground-truth bounding box covers both the visible and the (human-estimated) occluded region of the target object in every frame, as recommended by the VOT tracking challenge organisers [1].

3) Attributes annotation: "Attributes" define various challenging factors of tracking, and can provide a qualitative or functional interpretation of the characteristics of each sequence. To thoroughly analyse the dataset bias, we provide two categories of attributes annotations for each frame: binary and statistical, shown in Tab. I. The former are decided by human annotators and include background colour camouflage (BCC), background shape camouflages (BSC) and partial occlusion (PO). The latter are computed objectively from the statistical properties of features within the annotated ground-truth bounding boxes, including illumination variation (IV), depth variation (DV), scale variation (SV), colour distribution variation (CDV), depth distribution variation (DDV), surrounding depth clutter (SDC), and surrounding colour clutter (SCC).

TABLE I
THE DESCRIPTIONS OF ANNOTATED ATTRIBUTES (CS: COMPUTATIONAL STATISTICS; HA: HUMAN ANNOTATION.).

Attr.	Description	Annotation method
IV	Illumination variation – RGB intensity change of all pixels inside the bounding box (mean value).	CS
DV	Depth variation – depth change of all pixels inside the bounding box (mean value).	
SV	Scale variation – scale change of the bounding box (relatively ratio).	
CDV	Color distribution variation – RGB distribution change of the bounding box.	
DDV	Depth distribution variation – depth distribution change of the bounding box.	
SDC	Surrounding depth clutter – depth similarity between the target and ring-shaped contextual region (mean value).	
SCC	Surrounding color clutter – RGB intensity similarity between the target and ring-shaped contextual region (mean value).	
BCC	Background color camouflages – if the object in the background shares the similar color as the target (binary value).	HA
BSC	Background shape camouflages – if the object in the background shares the similar shape as the target (binary value).	
PO	Partial occlusion – if the target is partially occluded (binary value).	

B. Bias analysis

1) Tracking performance limits: To evaluate bias in the dataset [39], we first compute upper-limits and lower-limits that a tracker could achieve in each sequence, represented by overlap ratio between the bounding boxes of the tracker and the ground truth.

Upper-limits (green column heights in Fig. 6) represent ideal tracking performance (perfect match to ground truth). Since some tracking methods are not designed with scale adaptation, another upper-limit (red column heights in Fig 6) is computed for a tracker which matches the true target centroid position but with the initialized (first-frame) target scale.

[1]We released the code of our tracker, as well as the complete benchmark video dataset, on our web pages.

Lower-limits (blue columns in Fig. 6) represent the performance that can be achieved by a "dumb" tracker, defined as the best performing out of setting the bounding box: i) permanently in the image centre; ii) fixed permanently at the initialisation location in the first frame; iii) positioned randomly in each frame.

Sequences with high lower-limit should be completely avoided [39], since arbitrary (or "dumb") tracking algorithms tested on those sequences could achieve similarly good performance. Trackers without scale adaptation are best evaluated on sequences with high red column heights in Fig. 6, since those sequences have a larger range of possible performance results, and the performance scores of trackers will primarily be linked to bounding-box centre errors, and not to the tracker's capability for scale adaptation. Also, for those trackers without scale adaptation, comparing their results with respect to the maximum red column height is more objective and informative (compared to the 100% green column height). This is because no matter how accurate a tracker's position is, its tracking performance cannot exceed the red column height (upper-limit). In contrast, trackers with scale adaptation are better tested on sequences with low red column heights, so that the scale adapting abilities of the tracker will significantly influence the overall tracking performance metric.

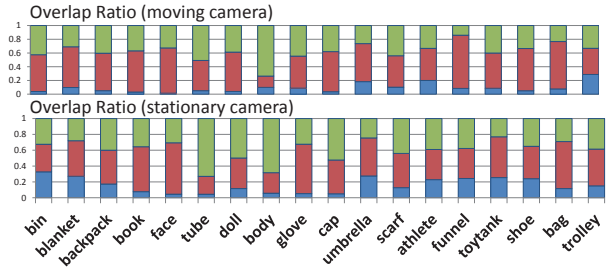


Fig. 6. Limits of tracking performance in each sequence. Blue, green and red column heights correspond to lower limits and upper limits with/without scale adaptation, respectively.

2) Attributes statistics: We first calculate the number of frames for each **binary attribute**, including background colour camouflage (BCC), background shape camouflages (BSC) and partial occlusion (PO), shown in Fig. 7. To avoid bias, the dataset should have a wide spread of attributes [20], shown as different colour sections in Fig. 7. It can be seen that less than half of the scenes have the attributes of BCC and BSC, but these attributes often last for many frames. In contrast, partial occlusion occurs in more scenes than BCC or BSC, but each instance lasts only for a smaller number of frames. This supports our assertion that all sequences should be per-frame annotated, since none of the annotated attributes last throughout any entire sequence. It is preferable to evaluate each tracker with respect to attributes annotated on a per-frame basis, for a more accurate and meaningful analysis.

For each sequence, we can also use the number of attributes per frame to evaluate the overall tracking difficulty. To appreciate how challenging our dataset is, as shown in Fig. 8, the majority of image frames in the dataset contain either one or two challenging attributes, and a few frames are extremely challenging (containing all binary attributes).

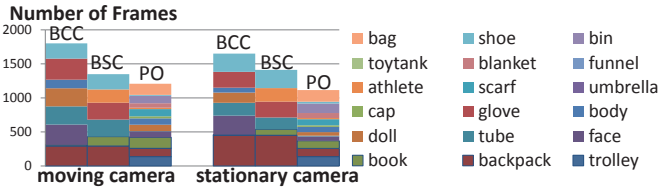


Fig. 7. Number of frames for each binary attribute. Different colour sections represent different sequences. BCC, BSC, PO are explained in Sec. IV-A.

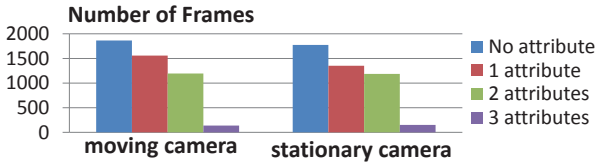


Fig. 8. Proportions of dataset frames which contain various numbers of challenging (binary) attributes.

Statistical attributes, such as IV, DV, SV, CDV, DDV, SCC and SDC are computed from all frames in all sequences. For IV and DV, we compute the mean value of all pixels inside the bounding box in terms of intensity and depth. The differences of those values between successive frames are used to represent the respective statistical attributes.

For CDV and DDV, we use the colour or depth values of all the pixels inside the ground-truth bounding box to compute distributions, which are then compared to the respective distributions in the next frame. The Bhattacharyya coefficient between successive distributions defines the CDV or DDV values for each frame.

SCC and SDC are defined as the similarity (according to Bhattacharyya coefficient) between the distributions of pixel values (colour or depth respectively) within the target bounding box region, and within a ring-shaped "context" region which surrounds the target's bounding box.

SV is defined as the scale change of the bounding box between successive frames. The changes of the target scale, between two consecutive frames, are normalized with respect to the absolute value of the bounding box scale in the first of the two frames.

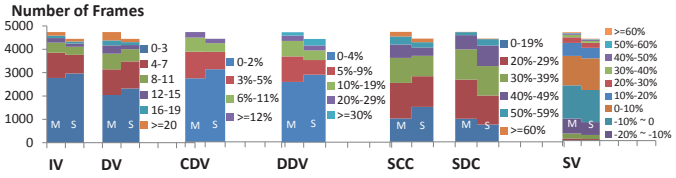


Fig. 9. Statistical attributes of our dataset. Colour sections denote different ranges of severity of each attribute - see right-hand legend for an explanation of each colour's meaning. IV, DV are measured by absolute value change (depth normalized 0 to 255). CDV, DDV, SBC, SCC are based on histogram while SV measures scale change ratio. **M** denotes moving camera and **S** denotes stationary camera.

Fig. 9 suggests that there is a good spread of different attributes, and each has a good spread of severities, over the frames of the dataset. These statistical attributes are useful to: 1) guide unbiased dataset construction; 2) provide functional

explanations of tracking failures; 3) help researchers choose optimal algorithm parameters for particular tracking conditions or applications.

V. Experimental results

We have tested our RGB-D tracker on our new dataset using two (currently competing) state-of-the-art evaluation methodologies. The OTB methodology [44] runs each tracker throughout each sequence, without reinitialising trackers following any tracking failures (i.e., a tracker which fails early will be ranked far lower than a tracker which fails later in the same sequence). In contrast, the VOT [1] methodology automatically re-initialises the tracker (using the ground-truth position in the following frame) whenever a tracking failure is detected.

We present the results of comparing our RGB-D tracker against the top four ranked RGB-D trackers [34] from recent literature (Princeton RGBD dataset): PT tracker [34], DS-KCF tracker [6], improved DS-KCF* tracker (with shape adaptation) [12], and OAPF tracker [28]. The test results are presented for i) the entire dataset; ii) stationary vs. moving camera; iii) frames corresponding to different kinds of attributes. Note that the trackers are tested on a laptop with a single core Intel I7-3720QM 2.60 Hz CPU and 8GB RAM.

A. Qualitative evaluation

The performance of the proposed and the compared trackers, in a variety of severely challenging scenes, is illustrated in Fig. 10. In sequence *Athlete*, the target deforms significantly and the range of the target from the camera also changes dramatically during the sequence. This requires the tracker to be robust to both shape and depth variations. As shown in Fig. 10, all four comparison methods fail at various points in the sequence (by fixating on non-target people), while our proposed algorithm successfully tracks throughout.

In the *Backpack* sequence, a red backpack is camouflaged while moving near to a red garbage bin. Our proposed tracker tracks accurately and robustly throughout the sequence. In contrast, DS-KCF, DS-KCF* and PT fail by fixating on non-target objects. OAPF retains some overlap with the target region at all frames, but achieves very poor accuracy by enlarging and stretching its bounding box to include various distracting non-target objects.

In the *Face* sequence, the tracked object is a human face which is partially occluded in various ways throughout the sequence. Again, our proposed tracker robustly and accurately tracks throughout, while all four comparison methods fail in various ways at various frames.

The sequence *Trolley* was captured in outdoors, where the quality of the images (especially the depth images) is affected by strong background illumination [2]. In the right-most frame of this sequence shown in Fig. 10, PT has erroneously flagged a false occlusion detection, while OAPF has erroneously fixated on the person's legs instead of the trolley. DS-KCF an DS-KCF* successfully retain a track on the target object, but with somewhat less good accuracy than our tracker. Our proposed method tracks robustly and accurately throughout the sequence.

Fig. 10. Visualization of results on sequences: *Athlete*, *Backpack*, *Face*, *Trolley*, showing extreme target deformations, extreme target range variation and severe camouflage situations. Our proposed tracker robustly and accurately tracks throughout all sequences. In contrast, all four comparison methods fail in the first three sequences (by fixating on non-target objects). PT (false occlusion detection) and OAPF (fixation on non-target object) fail in the fourth sequence, while DS-KCF and DS-KCF* maintain the true target within its bounding box, but with somewhat poorer accuracy than our proposed method.

B. Quantitative evaluation

1) The effectiveness of the components of the proposed tracker: To evaluate the effectiveness of the proposed tracking components, we first decompose the full algorithm into separate components and measure the sensitivity of the tracking performance with respect to some key parameters. First, we include in our testing our previous RGB tracker, CDT [47] to demonstrate the effectiveness of depth features. Then, we define a baseline tracker "G" which only uses global features (as proposed in this paper). The algorithm denoted as "G+λ*ST+LC" includes the proposed global spatio-temporal (ST) consistency constraints with local-colour (LC) feature, where λ is the key parameter defined in Eq. 1. "GC" represents the global-colour feature while "GD" denotes the global-depth feature. "G+λ*ST+LC+GC" represents the full algorithm with different parameter λ settings. The results are shown as trade-off curves, where the x-axis is the threshold of the overlap ratio and y-axis is the success ratio with respect to the defined overlap ratio. The area under curves (AUC) is given inside the figure legend in Fig. 11.

Fig. 11 shows that CDT [47] performs better than the baseline tracker (G), which only uses the global features. Its better performance can be attributed to the local parts used in CDT. The trackers which use the global spatio-temporal (ST) consistency constraints and local features, i.e., LC, GC, GD, perform significantly better. The key parameter λ indeed affects tracking performance, but within an acceptable stable range.

2) Overall evaluation - OTB protocol: For evaluation following the OTB [44] protocol, there is no re-initialisation after tracking failures. Wu et al. [44] suggest to use *area under curve (AUC)* of either the overlap ratio, or centre-

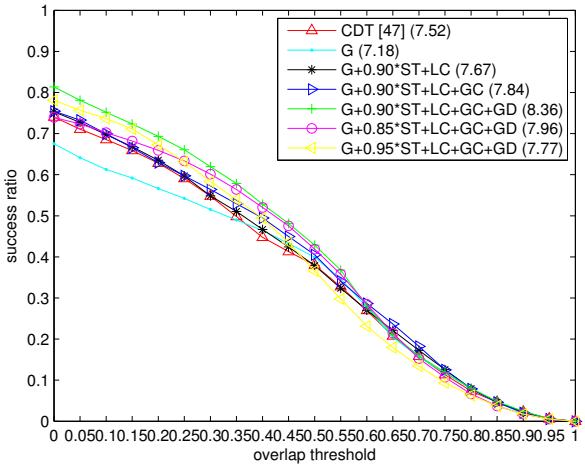


Fig. 11. The trade-off curves with different configurations of the tracking components.

error distance, for performance evaluation. The centre-error measure (the distance between the tracker and the ground-truth centroids) includes all frames throughout the entire sequence, even after total tracking failure which may occur early in the sequence. Čehovin at al. [8] argue that centre-error measures are sensitive to subjective human bounding box annotation, which can be significantly variable and a poor reflection of the true tracking performance. Therefore, in this paper, we compare each tracker in terms of the AUC of the bounding-box overlap ratio (i.e., % overlap between the ground-truth target bounding box and that estimated by the tracker).

Tab. II shows the AUC performance of each tracker. Our proposed tracker clearly demonstrates the best overall performance. It also shows the best performance within most of the attribute sub-sets, i.e., SV, CDV, DDV, SDC, BCC, BSC. DS-KCF* [12] ranks second overall, which improves on its predecessor DS-KCF [6] with a large margin. PT [6] ranks third while OAPF ranks at the bottom according to the OTB protocol. With the proposed model, our tracker (matlab version) achieves quite competitive processing speed, shown in Tab. II.

To gain a deeper understanding about how the target scale affects the processing time, we further investigate the frame-per-second (fps) with respect to the target scale, as shown in Fig. 12. The x-axis represents the average scale of the target, counted by the number of the pixels, in every sequence. The y-axis is the corresponding fps. One can see that the computational burden increases linearly with respect to the target scale.

3) Overall evaluation - VOT protocol: The evaluation following the VOT protocol [1] is based on two independent metrics: *accuracy* (overlap ratio between the tracker output and the ground truth bounding boxes); and *robustness*, measured by the frequency of complete tracking failures (when the overlap ratio becomes zero). Whenever failures occur, the tracker is automatically reinitialised (using the ground-truth target position in the following frame) to continue tracking.

Failure rates and accuracy scores are shown in Tab. III. Our

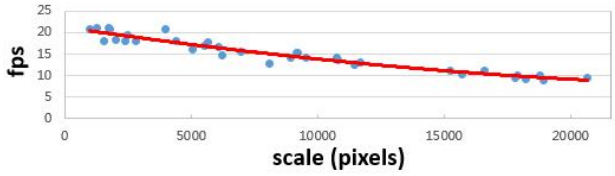


Fig. 12. Fps with respect to the target scale. The blue dots are the original data computed from every sequence. The red line is the fitted curve.

proposed tracker ranks first in robustness, with competitive accuracy, while DS-KCF* [12] ranks second in robustness. Note that accuracy scores can be misleading. In VOT, since ground truth is used to re-initialise trackers (with perfect accuracy) following every tracking failure, less robust trackers (many failures) are likely to exhibit *artificially higher accuracy* scores. Therefore, the accuracy metric *is only meaningful when comparing trackers which have the same robustness score*. Our method's robustness score significantly exceeds all others, which means that it is not meaningful to compare our accuracy against that of other methods. Nevertheless, even with our greater robustness scores, our accuracy scores still consistently outperform those of DS-KCF*, OAPF and are competitive with those of DS-KCF and PT. Meanwhile, PT [34] exhibits a very large number of total tracking failures (suggesting that its comparatively high accuracy score is merely an artefact of the post-failure re-initialisations as explained above).

4) Overall evaluation - OTB vs VOT: Our tracker consistently performs best, on both OTB and VOT evaluation protocols. Note that the performance of PT [34] changes dramatically between protocols (worst in VOT but third in OTB). This is because PT often loses the target but then successfully re-captures it later, while VOT anyway re-initialises all trackers immediately following failures. Note that in OTB, success ratio can dramatically be affected by the sequential positions of failures. If a tracker fails early in a sequence, it will show a low success ratio, but if it fails late in a sequence, it may show high success ratio. Therefore, recognising that neither methodology is perfect, we have evaluated our tracker using both VOT and OTB methodologies, and demonstrated superior performance in both.

5) Test on Princeton dataset: In Sec. I-C we have already discussed some of the problems with the RGB-D tracking dataset PTB [34], however, since it has widely been used in the RGB-D tracking community, we will, leaving the problematic nature of the dataset aside, still test our tracker on this dataset. In terms of target type, we achieve 65 (Princeton defined evaluation score) on *human* category (better than OAPF), 67 on *animal* category (better than 3D-T), 74 on *rigid target* (better than DS-KCF*). In terms of target size, movement, occlusion and motion type, our performance is very comparable to DS-KCF. We achieve 68 on *large target*, 69 on *small target*, 72 on *slow target*, 68 on *fast target*, 61 on *occlusion*, 80 on *non-occlusion*, 78 on *passive motion type* and 66 on *active motion type*. Despite the aforemention problems with the dataset, our proposed tracker still achieves comparable performance (overall better than DS-KCF c++ version). Note that the other trackers, i.e., OAPF, DS-KCF, DS-KCF*, PT,

TABLE II
AUC OF BOUNDING-BOX OVERLAP. RED DENOTES BEST PERFORMING TRACKER. BLUE DENOTES SECOND BEST.

	Overall	Stationary	Moving	IV	DV	SV	CDV	DDV	SDC	SCC	BCC	BSC	PO	fps
Ours	8.36	9.57	7.18	5.78	7.56	5.07	5.12	7.66	8.01	9.53	6.67	7.17	7.73	14.73
DS-KCF* [12]	8.21	9.36	7.13	6.10	7.88	4.39	0.94	5.26	7.93	9.81	5.66	6.50	7.76	17.89
PT [34]	7.44	8.54	6.43	4.15	6.65	2.80	0.43	3.48	6.79	8.23	5.74	5.76	6.20	0.14
DS-KCF [6]	7.23	7.52	6.85	5.50	7.06	3.36	1.43	4.16	7.89	8.25	4.82	5.16	6.02	20.95
OAPF [28]	5.24	6.0	4.54	3.19	4.45	3.07	3.22	3.71	5.00	6.13	3.79	4.82	5.82	1.30

TABLE III
VOT PROTOCOL RESULTS. RED DENOTES BEST PERFORMING TRACKER. BLUE DENOTES SECOND BEST.

		Overall	Stat.	Moving	IV	DV	SV	CDV	DDV	SDC	SCC	BCC	BSC	PO
Ours	Fail.	2.11	1.56	2.67	0.11	0.50	0.19	0.00	0.25	0.97	0.22	1.17	0.67	0.50
	Acc.	0.52	0.54	0.50	0.41	0.49	0.41	0.46	0.47	0.52	0.52	0.49	0.50	0.46
DS-KCF* [12]	Fail.	2.69	2.50	2.89	0.06	0.78	0.22	0.03	0.36	1.44	0.69	1.67	1.28	0.58
	Acc.	0.50	0.51	0.50	0.48	0.49	0.46	0.41	0.53	0.44	0.43	0.57	0.57	0.49
PT [34]	Fail.	10.61	8.61	12.61	0.33	3.14	1.42	0.19	1.03	5.78	2.33	4.31	3.83	3.61
	Acc.	0.64	0.63	0.64	0.53	0.65	0.55	0.60	0.65	0.62	0.62	0.64	0.63	0.57
DS-KCF [6]	Fail.	3.36	3.00	3.72	0.08	0.64	0.28	0.03	0.31	1.39	0.92	1.56	1.47	1.11
	Acc.	0.56	0.56	0.47	0.50	0.56	0.48	0.54	0.58	0.55	0.52	0.53	0.51	0.54
OAPF [28]	Fail.	5.78	3.72	7.83	0.19	1.42	0.50	0.11	0.61	2.75	1.36	2.61	1.86	1.47
	Acc.	0.35	0.37	0.33	0.25	0.32	0.25	0.22	0.27	0.35	0.36	0.32	0.35	0.33

utilize RGB features and depth features in a decoupled way, e.g., RGB for tracking and depth for scale adaptation or occlusion reasoning, while our tracker continuously checks the ambiguous situations using RGB and depth features in the global layer (Sec. III-A), and jointly utilises RGB and depth features in the clustered decision tree (coupled way) in the local layer (Sec. III-B). It is obvious that if the RGB images and depth images are not well synchronised, such an effect causes more severe problems for a tracker, which uses depth and RGB features in a coupled way. Therefore, the effect of the unsynchronized image pairs (RGB and depth) on the aforementioned trackers is smaller than that on our tracker. This explains why the performance of our tracker dropped on Princeton RGB-D dataset.

VI. CONCLUSION

This paper has presented a novel and strongly performing method for RGB-D target tracking, and also presented a new benchmark dataset. The proposed tracker incorporates a hierarchical two-layered target model, and exploits spatio-temporal consistency constraints of both colour and depth information. In the global layer, the algorithm exploits temporal consistency to adaptively fuse features for candidate region searching. The global candidate region is then split into a set of local candidate regions for robust matching to local target parts. Parts matching is robustified by considering both global feature statistics and contextual information. A spatial constraint is extracted from the depth context to accurately estimate an adaptive target depth interval, leading to an effective range-invariant model of the target's depth data. For evaluation, we developed a new RGB-D benchmark dataset with *per-frame* annotated attributes and extensive bias analysis. Our tracker was evaluated using two different state-of-the-art methodologies [1], [44]. According to both evaluation methodologies, our tracker demonstrated superior performance over four currently top ranked, state-of-the-art RGB-D tracking methods.

ACKNOWLEDGEMENT

This work was funded by EU H2020 RoMaNS 645582. It was also partially supported by EPSRC EP/M026477/1 and the National Key Research and Development Program of China, No. 2016YFC0103100. We also acknowledge MoD/Dstl and EPSRC for providing the grant to support the UK academics (Aleš Leonardis) involvement in a Department of Defense funded MURI project. We also want to thank for the great help from Heng Yang for the insightful discussions.

REFERENCES

[1] The VOT chanllenge. http://www.votchallenge.net/.
[2] S. M. Abbas and A. Muhammad. Outdoor RGB-D slam performance in slow mine detection. In *7th German Conference on Robotics*, pages 1–6, 2012.
[3] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012.
[4] A. K. Bhattacharya. On a measure of divergence between two statistical populations defined by their probability distributions. In *Bulletin of the Calcutta Mathematical Society, issue 35*, pages 99–110, 1943.
[5] A. Bibi, T. Zhang, and B. Ghanem. 3d part-based sparse tracker with automatic synchronization and registration. In *CVPR*, pages 1439–1448, 2016.
[6] M. Camplani, S. Hannuna, M. Mirmehdi, D. Damen, A. Paiement, L. Tao, and T. Burghardt. Real-time RGB-D tracking with depth scaling kernelised correlation filters and occlusion handling. In *BMVC*, 2015.
[7] L. Cao and R. Ji. Robust depth-based object tracking from a moving binocular camera. *Signal Processing*, pages 154–161, 2015.
[8] L. Čehovin, A. Leonardis, and M. Kristan. Visual object tracking performance measures revisited. *TIP*, 25(3):1261–1274, 2016.
[9] Y. Chen, Y. Shen, X. Liu, and B. Zhong. 3D object tracking via image sets and depth-based occlusion detection. *Signal Processing*, pages 146–153, 2015.
[10] M. Danelljan, F. S. Khan, M. Felsberg, and J. v. d. Weijer. Adaptive colour attributes for real-time visual tracking. In *CVPR*, pages 1090–1097, 2014.
[11] X. Dong and M. J. Chantler. Perceptually motivated image features using contours. *TIP*, 2016.
[12] S. Hannuna, M. Camplani, J. Hall, M. Mirmehdi, D. Damen, T. Burghardt, A. Paiement, and L. Tao. Ds-kcf: a real-time tracker for rgb-d data. *Journal of Real-Time Image Processing*, pages 1–20, 2016.
[13] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *PAMI*, 2015.
[14] Z. Hong, X. Mei, and D. Tao. Dual-force metric learning for robust distracter-resistant tracker. In *ECCV*, pages 513–527, 2012.

[15] W. Hu, W. Li, X. Zhang, and S. Maybank. Single and multiple object tracking using a multi-feature joint sparse representation. *PAMI*, 37(4):816–833, 2015.
[16] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *PAMI*, 34(12):2420–2440, 2012.
[17] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, pages 1822–1829, 2012.
[18] N. Jiang and Y. Wu. Unifying spatial and attribute selection for distracter-resilient tracking. In *CVPR*, pages 3502–3509, 2014.
[19] M. Kristan, S. Kovacic, A. Leonardis, and J. Pers. A two-stage dynamic model for visual tracking. *SMC*, 40(6), 2010.
[20] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, and L. Cehovin. A novel performance evaluation methodology for single-target trackers. *PAMI*, 2015.
[21] M. Kristan et al. The visual object tracking challenge results. In *ICCVW*, pages 98–111, Dec 2013.
[22] J. Kwon and K. M. Lee. Highly nonrigid object tracking via patch-based dynamic appearance modeling. *PAMI*, 35(10):2427–2441, 2013.
[23] K. Lebeda, S. Hadfield, and R. Bowden. Exploring causal relationships in visual object tracking. In *ICCV*, 2015.
[24] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan. Nus-pro: A new visual tracking challenge. *PAMI*, PP(99), 2015.
[25] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo. Robust visual tracking based on incremental tensor subspace learning. In *ICCV*, pages 1–8, 2007.
[26] M. Luber, L. Spinello, and K. O. Arras. People tracking in rgb-d data with on-line boosted target models. In *IROS*, pages 3844–3849, 2011.
[27] X. Mei and H. Ling. Robust visual tracking using L1 minimization. In *ICCV*, pages 1436–1443, 2009.
[28] K. Meshgia, S.-i. Maedaa, S. Obaa, H. Skibbea, Y.-z. Lia, and S. Ishiia. Occlusion aware particle filter tracker to handle complex and persistent occlusions. *CVIU*, 2015.
[29] M. Munaro, F. Basso, and E. Menegatti. Tracking people within groups with RGB-D data. In *IROS*, pages 2101–2107, Oct 2012.
[30] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003.
[31] F. Pernici and A. D. Bimbo. Object tracking by oversampling local features. *PAMI*, 99:1, 2013.
[32] H. Possegger, T. Mauthner, and H. Bischof. In defense of color-based model-free tracking. In *CVPR*, pages 2113–2120, 2015.
[33] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah. Visual tracking: an experimental survey. *PAMI*, 36(7):1442–1468, 2014.
[34] S. Song and J. Xiao. Tracking revisited using RGB-D camera: Unified benchmark and baselines. In *ICCV*, pages 233–240, 2013.
[35] R. Stolkin, I. Florescu, and G. Kamberov. An adaptive background model for camshift tracking with a moving camera. In *International Conference on Advances in Pattern Recognition*, pages 147–151, 2007.
[36] R. Stolkin, A. Greig, and J. Gilby. A calibration system for measuring 3d ground truth for validation and error analysis of robot vision algorithms. *Measurement Science and Technology*, 17(10):2721, 2006.
[37] R. Stolkin, A. Greig, and J. Gilbyc. Measuring complete ground-truth data and error estimates for real video sequences, for performance evaluation of tracking, camera pose and motion estimation algorithms. In *ICCV workshop*, pages 25–30, 2005.
[38] M. Talha and R. Stolkin. Particle filter tracking of camouflaged targets by adaptive fusion of thermal and visible spectra camera data. *IEEE Sensors Journal*, 14(1):159–166, 2014.
[39] A. Torralba, A. Efros, et al. Unbiased look at dataset bias. In *CVPR*, pages 1521–1528, 2011.
[40] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *TIP*, 18(7):1512–1523, 2009.
[41] L. Čehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *PAMI*, Apr 2013.
[42] T. Vojir and J. Matas. Online adaptive hidden markov model for multi-tracker fusion. *CVIU*, 2016.
[43] L. Wang, T. Liu, G. Wang, K. L. Chan, and Q. Yang. Video tracking using learned hierarchical features. *TIP*, 24(4):1424–1435, 2015.
[44] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, pages 2411–2418, 2013.
[45] Y. Wu, J. Lim, and M.-H. Yang. Object tracking benchmark. *PAMI*, 2015.
[46] J. Xiao, L. Qiao, R. Stolkin, and A. Leonardis. Distractor-supported single target tracking in extremely cluttered scenes. In *ECCV*, 2016.
[47] J. Xiao, R. Stolkin, and A. Leonardis. Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In *CVPR*, 2015.
[48] J. Xiao, R. Stolkin, M. Oussalah, and A. Leonardis. Continuously adaptive data fusion and model relearning for particle filter tracking with multiple features. *IEEE Sensors Journal*, 16(8):2639–2649, 2016.
[49] T. Zhang, B. Ghanem, S. Liu, C. Xu, and N. Ahuja. Robust visual tracking via exclusive context modeling. *IEEE transactions on cybernetics*, 46(1):51–63, 2016.
[50] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang. Structural sparse tracking. In *CVPR*, pages 150–158, June 2015.
[51] B. Zhong, Y. Shen, Y. Chen, W. Xie, Z. Cui, H. Zhang, D. Chen, T. Wang, X. Liu, S. Peng, et al. Online learning 3D context for robust visual tracking. *Neurocomputing*, 151:710–718, 2015.
[52] W. Zhong, H. Lu, and M.-H. Yang. Robust object tracking via sparsity-based collaborative model. In *CVPR*, pages 1838–1845, 2012.

Jingjing Xiao received her Bachelor and Master degree from College of Mechatronics Engineering and Automation, National University of Defence Technology, China, in 2010 and 2012, respectively. She holds a PhD degree from the University of Birmingham in 2016. Currently, she is an honorary research fellow at the University of Birmingham, U.K., and a full time researcher in Xinqiao Hospital, Third Military Medical University, Chongqing, China. Her research interests include object tracking with computer vision.

Rustam Stolkin currently serve as the Senior Birmingham Fellow in Robotics, based in the Department of Mechanical Engineering at University of Birmingham, where he has worked since 2008. His training included undergraduate and masters degrees in Engineering from Oxford University, and a PhD in Robot Vision undertaken between University College London and UK imaging industry. He also worked as a Assistant Professor (Research) at Stevens Institute of Technology, USA, 2004-2008. His main interests include vision and sensing, robotic grasping and manipulation, robotic vehicles, human-robot interaction, AI and machine learning.

Yuqing Gao received the BEng degrees in Computer Science and Technology from Harbin Institute of Technology (HIT), China, and in Computer System Engineering from University of Birmingham, UK. She is now doing her PhD in Electronic, Electrical and Systems Engineering at the University of Birmingham. Her research topics are about vision-based augmented reality system and the related image processing technologies.

Aleš Leonardis is a Professor at the School of Computer Science, University of Birmingham and co-Director of the Centre for Computational Neuroscience and Cognitive Robotics. He is also a Professor at the FCIS, University of Ljubljana and adjunct professor at the FCS, TU-Graz. His research interests include robust and adaptive methods for computer vision, object and scene recognition and categorization, statistical visual learning, 3D object modeling, and biologically motivated vision.